Pemrograman Berorientasi Objek dengan C#

[Lecture Note Pertemuan ke - 09]
(Objek Kontrol / Lanjutan 1)

Ng Poi Wong, 2020, Sesi 4.c : Objek Kontrol (Lanjutan 1), Lecture Notes, Pemrograman Berorientasi Objek dengan C# (IF0058), STMIK Mikroskil Medan, Dikirimkan 02 Maret 2020.

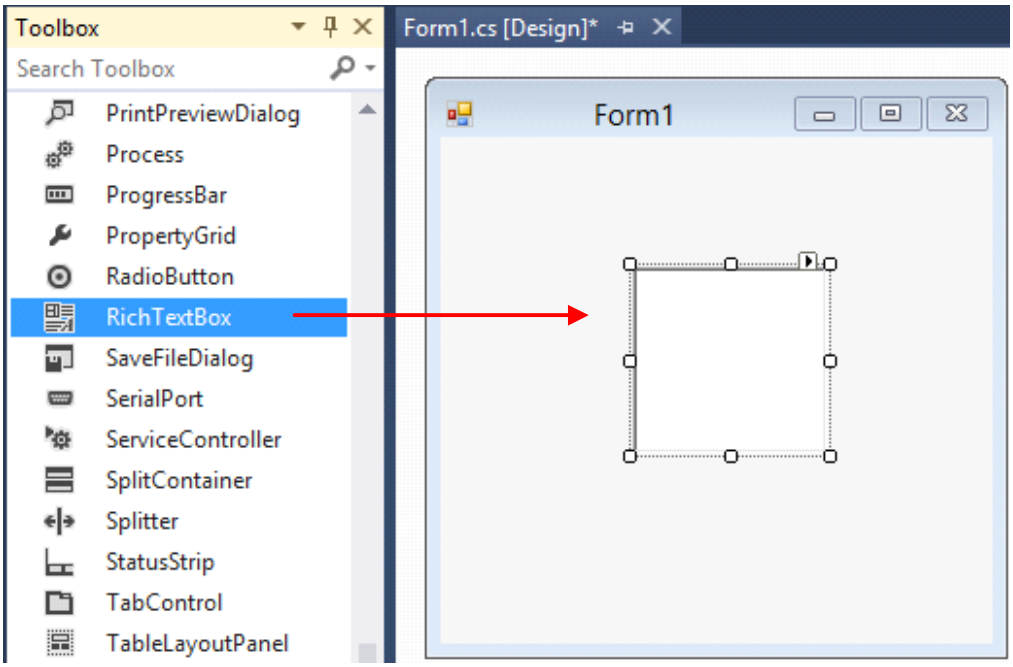
Capaian MK : Mahasiswa mampu memanfaatkan kontrol objek dasar dan lanjutan pada form

DAFTAR ISI PENJELASAN SLIDE

Slide 03 s/d 07	**[RichTextBox]**	**2**
Contoh - 1	**[Contoh Kasus Sederhana - 1]**	**5**
Slide 08 s/d 19	**[ListView]**	**9**
Contoh - 2	**[Contoh Kasus Sederhana - 2]**	**14**

PENJELASAN DARI SLIDE ke-03 s/d 07

1. Objek **RichTextBox** adalah sama dengan objek **TextBox**, dimana terdapat tambahan sejumlah fitur untuk mengelola teks yang diinput oleh user.



2. Secara umum, penggunaan objek **TextBox** identik dengan menggunakan aplikasi **NotePad** pada OS Windows, dan penggunaan objek **RichTextBox** adalah identik dengan menggunakan aplikasi **WordPad** pada OS Windows, dimana aplikasi NotePad & WordPad adalah aplikasi bawaan dari OS Windows.
3. Objek **RichTextBox** dapat baca/tulis file dokumen berekstensi **.RTF** (Rich Text File), dimana file berekstensi tersebut langsung dikenali oleh aplikasi word prosessor seperti Microsoft Word.
4. Untuk mempermudah pengenalan terhadap objek **RichTextBox** yang digunakan, umumnya seorang programmer memberikan penyingkatan nama terhadap **RichTextBox** tersebut dengan penyingkatan nama "**rtb_[nama]**", misalnya "**rtbDokumen**", "**rtbDaftar**", dan sebagainya.
5. Berikut penjelasan beberapa **Properties** yang umum digunakan pada **RichTextBox** :

Properties	Keterangan
AcceptsTab	Mengatur penerimaan karakter Tab (Tabulasi) ke dalam objek. Bernilai True atau False (Default = False).
DetectUrls	Mengatur pendeteksian suatu teks di dalam objek adalah URL atau bukan. Bernilai True atau False (Default = True).
Lines	Mengambil isi tulisan dari objek secara per baris dalam satuan Array.
ScrollBars	Mengatur kemunculan scrollbar pada objek. Terdiri dari None, Horizontal, Vertical, Both, ForcedHorizontal, ForcedVertical, ForcedBoth (Default = Both).

PENJELASAN DARI SLIDE ke-03 s/d 07

Properties	Keterangan
WordWarp	Mengatur fitur penyesuaian teks dengan lebar dari objek. Bernilai True atau False (Default = True).
ZoomFactor	Mengatur skala zooming terhadap teks pada objek. Bernilai minimum 1/64 (0,015625) dan maksimal 64 (Default = 1).

6. Berikut penjelasan beberapa **Method** yang umum digunakan pada **RichTextBox** :

Method	Keterangan
AppendText	Untuk menambahkan suatu tulisan pada posisi akhir dari isi tulisan pada objek.
CanRedo	Untuk mengecek apakah dapat dilakukan Redo atau tidak.
CanUndo	Untuk mengecek apakah dapat dilakukan Undo atau tidak.
ClearUndo	Untuk mengosongkan memori dari penyimpanan perintah Undo.
Copy	Untuk menjalankan proses Copy terhadap teks yang terpilih pada objek ke dalam memori.
Cut	Untuk menjalankan proses Cut terhadap teks yang terpilih pada objek ke dalam memori.
LoadFile	Untuk memanggil file dokumen berekstensi .RTF
Paste	Untuk menjalankan proses Paste dari memori ke dalam objek.
Redo	Kebalikan dari Undo.
SaveFile	Untuk menyimpan isi objek menjadi file dokumen berekstensi .RTF
SelectAll	Untuk memilih semua teks yang ada pada objek.
SelectedText	Untuk mengambil teks yang terpilih (ter-highlight).
SelectionAlignment	Untuk memberikan perataan terhadap teks yang terpilih. Bertipe HorizontalAlignment, dan terdiri dari pilihan Center, Left, dan Right.
SelectionBackColor	Untuk mengatur warna latar belakang dari teks yang terpilih.
SelectionBullet	Untuk memberikan penomoran bullet pada teks yang terpilih. Bernilai True atau False.
SelectionCharOffset	Untuk mengatur teks yang terpilih menjadi superscript atau subscript. Diberikan nilai positif untuk superscript, dan nilai negatif untuk subscript.
SelectionColor	Untuk mengatur warna tulisan dari teks yang terpilih.
SelectionFont	Untuk mengatur Font dari teks yang terpilih.
SelectionLength	Untuk mengatur/mengambil panjang teks dari teks yang terpilih.
SelectionProtected	Untuk mengunci teks yang terpilih sehingga tidak bisa diedit.

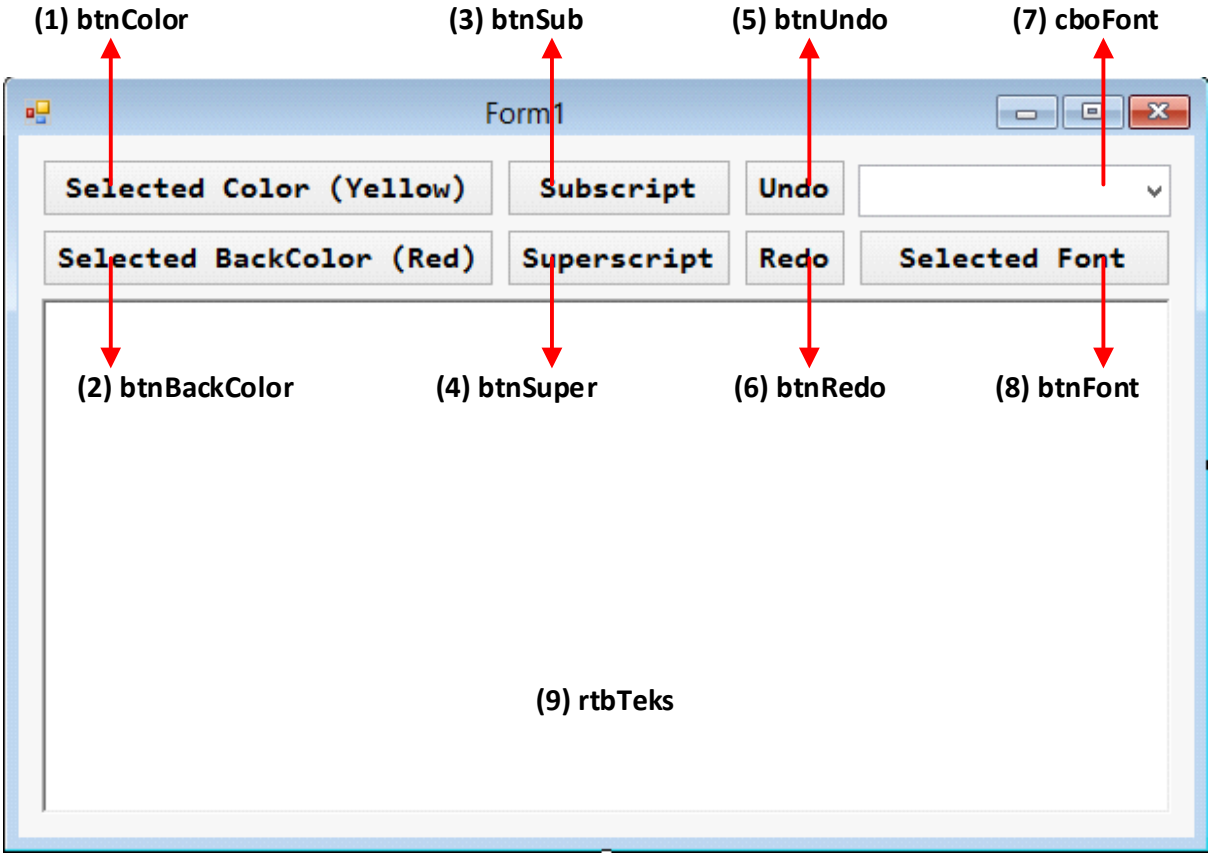
PENJELASAN DARI SLIDE ke-03 s/d 07

Method	Keterangan
SelectionStart	Untuk mengatur/mengambil posisi indeks dari karakter pertama dari teks yang terpilih.
Undo	Untuk mengulang perintah yang sudah dibatalkan sebelumnya.

CONTOH KASUS SEDERHANA - 1

Untuk contoh kasus sederhana dari penggunaan objek **RichTextBox**, dapat dilihat contoh berikut :

1. Rancang tampilan Windows Form Application sebagai berikut :



2. Untuk masing-masing objek dari rancangan tampilan di atas, memiliki keterangan **Properties** sebagai berikut :

   a. Untuk **Properties** dari **Form** :

Properties	Keterangan
Font	Name = Consolas Size = 12 Style = Bold

   b. Untuk **Properties** dari **Button** "**btnColor**" **(1)** :

Properties	Keterangan
Name	"btnColor"
Text	"Selected Color (Yellow)"

CONTOH KASUS SEDERHANA - 1

c. Untuk **Properties** dari **Button** "**btnBackColor**" **(2)** :

Properties	Keterangan
Name	"btnBackColor"
Text	"Selected BackColor (Red)"

d. Untuk **Properties** dari **Button** "**btnSub**" **(3)** :

Properties	Keterangan
Name	"btnSub"
Text	"Subscript"

e. Untuk **Properties** dari **Button** "**btnSuper**" **(4)** :

Properties	Keterangan
Name	"btnSuper"
Text	"Superscript"

f. Untuk **Properties** dari **Button** "**btnUndo**" **(5)** :

Properties	Keterangan
Name	"btnUndo"
Text	"Undo"

g. Untuk **Properties** dari **Button** "**btnRedo**" **(6)** :

Properties	Keterangan
Name	"btnRedo"
Text	"Redo"

h. Untuk **Properties** dari **ComboBox** "**cboFont**" **(7)** :

Properties	Keterangan
Name	"cboFont"

i. Untuk **Properties** dari **Button** "**btnFont**" **(8)** :

Properties	Keterangan
Name	"btnFont"
Text	"Selected Font"

j. Untuk **Properties** dari **RichTextBox** "**rtbTeks**" **(9)** :

Properties	Keterangan
Name	"rtbTeks"

CONTOH KASUS SEDERHANA - 1

3. Untuk masing-masing objek dari rancangan tampilan di atas, memiliki **Event** dengan kode program sebagai berikut :
   a. **Event** "**Load**" dari **Form** :

```
cboFont.Items.Add("Arial Black");
cboFont.Items.Add("Book Antiqua");
cboFont.Items.Add("Comic Sans MS");
cboFont.Items.Add("Consolas");
cboFont.Items.Add("Garamond");
cboFont.Items.Add("Impact");
cboFont.Items.Add("Lucida Calligraphy");
cboFont.Items.Add("Lucida Handwriting");
cboFont.Items.Add("MS Gothic");
cboFont.Items.Add("Trebuchet MS");
cboFont.Items.Add("Wingdings");
cboFont.Items.Add("Wingdings 2");
cboFont.Items.Add("Wingdings 3");
cboFont.DropDownStyle = ComboBoxStyle.DropDownList;
cboFont.SelectedIndex = 3;
```

   Penjelasan untuk kode program di atas adalah :
   Memasukkan beberapa nama Font ke dalam **ComboBox** "**cboFont**", kemudian mengubah properties **DropDownStyle** dari **ComboBox** "**cboFont**" dan menset font default adalah "Consolas".

   b. **Event** "**Click**" dari **Button** "**btnColor**" :

```
rtbTeks.SelectionColor = Color.Yellow;
```

   Penjelasan untuk kode program di atas adalah :
   Mengubah warna tulisan dari nilai teks yang terpilih (highlight) dari **RichTextBox** "**rtbTeks**".

   c. **Event** "**Click**" dari **Button** "**btnBackColor**" :

```
rtbTeks.SelectionBackColor = Color.Red;
```

   Penjelasan untuk kode program di atas adalah :
   Mengubah warna latar dari nilai teks yang terpilih (highlight) dari **RichTextBox** "**rtbTeks**".

   d. **Event** "**Click**" dari **Button** "**btnSub**" :

```
rtbTeks.SelectionCharOffset = -8;
```

   Penjelasan untuk kode program di atas adalah :
   Mengubah nilai teks yang terpilih (highlight) dari **RichTextBox** "**rtbTeks**" menjadi subscript dengan penurunan posisi sebesar 8 piksel.

   e. **Event** "**Click**" dari **Button** "**btnSuper**" :

```
rtbTeks.SelectionCharOffset = 8;
```

   Penjelasan untuk kode program di atas adalah :
   Mengubah nilai teks yang terpilih (highlight) dari **RichTextBox** "**rtbTeks**" menjadi superscript dengan kenaikan posisi sebesar 8 piksel.

CONTOH KASUS SEDERHANA - 1

f. **Event** "**Click**" dari **Button** "**btnUndo**" :

```
if (rtbTeks.CanUndo)
   rtbTeks.Undo();
```

Penjelasan untuk kode program di atas adalah :

Jika dimungkinkan dilakukan proses Undo, maka lakukan proses Undo.

g. **Event** "**Click**" dari **Button** "**btnRedo**" :

```
if (rtbTeks.CanRedo)
   rtbTeks.Redo();
```

Penjelasan untuk kode program di atas adalah :

Jika dimungkinkan dilakukan proses Redo, maka lakukan proses Redo.

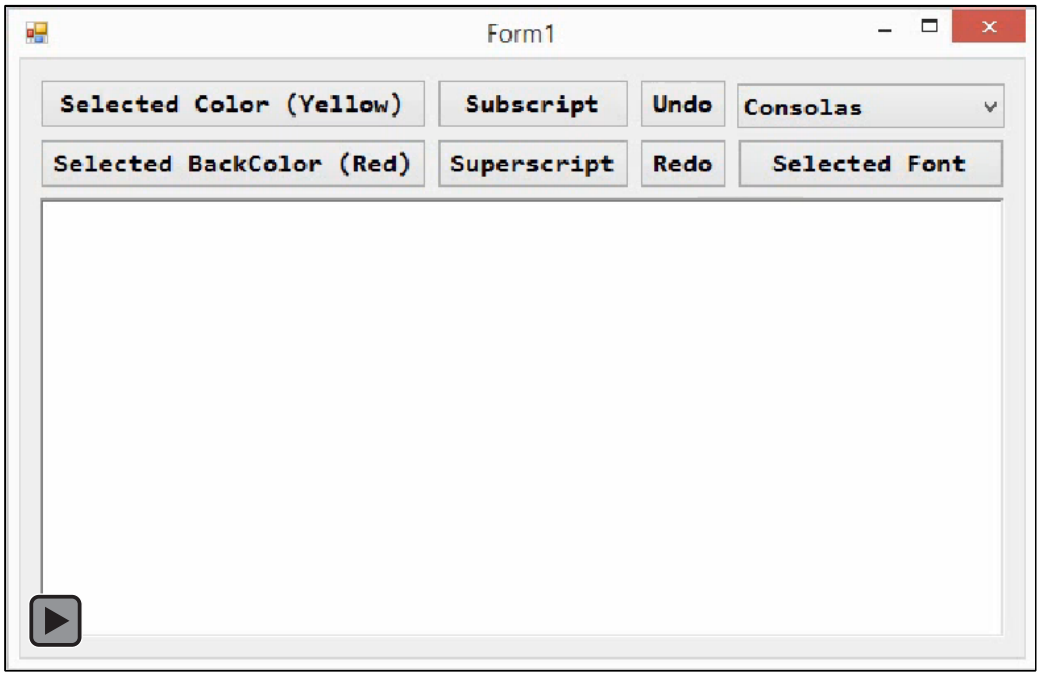
h. **Event** "**Click**" dari **Button** "**btnFont**" :

```
rtbTeks.SelectionFont = new Font(
   cboFont.Text, rtbTeks.Font.Size, rtbTeks.Font.Style);
```

Penjelasan untuk kode program di atas adalah :

Mengubah nama font dari nilai teks yang terpilih (highlight) dari **RichTextBox** "**rtbTeks**" menjadi nama font yang terpilih dari **ComboBox** "**cboFont**", tetapi untuk ukuran & style dari font tetap mengambil status ukuran & style yang sedang digunakan oleh **RichTextBox** "**rtbTeks**" sendiri.

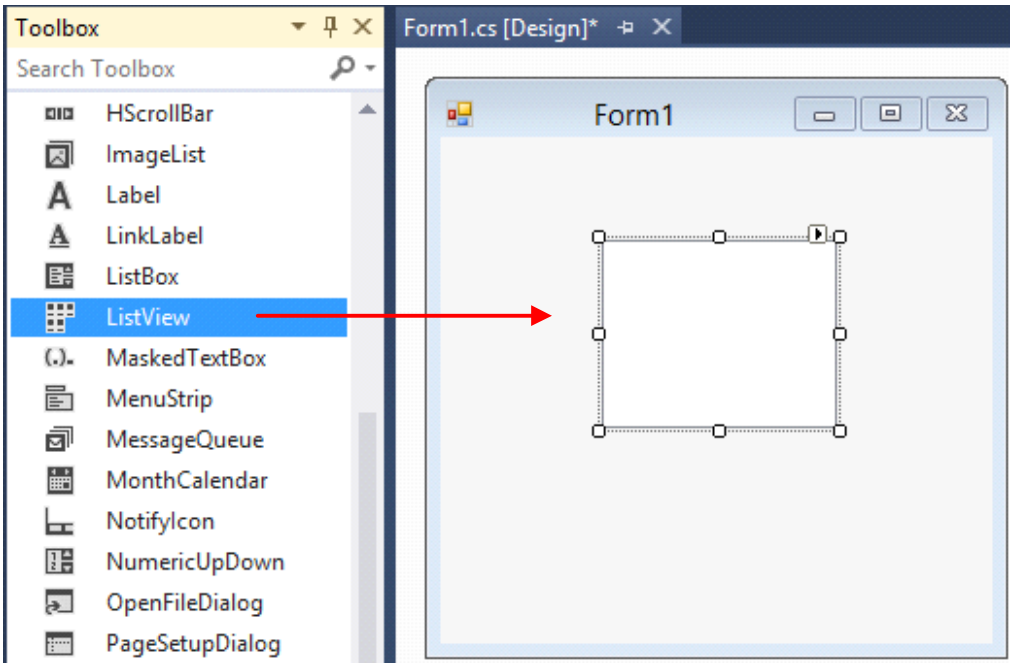
4. Dari contoh kasus sederhana ini, jika setelah selesai dirancang dan dituliskan kode program untuk **Event** yang ada di atas, maka akan memiliki tampilan video sebagai berikut :



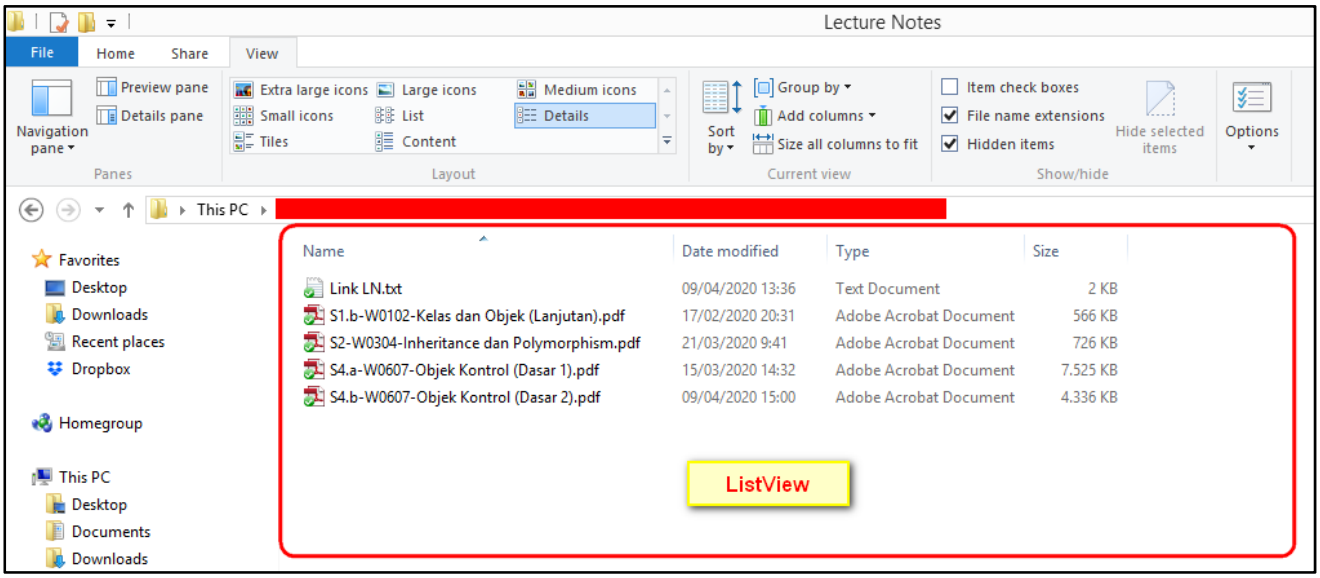
Note : Memerlukan Adobe Flash Player untuk dapat memutar video di atas.

PENJELASAN DARI SLIDE ke-08 s/d 19

7. Objek **ListView** berfungsi sama dengan ListBox, dimana menampilkan nilai dalam bentuk Daftar (List), tetapi dengan tambahan sejumlah fitur untuk mengelola daftar yang disajikan.



8. Objek ListView juga sering dijumpai pada OS Windows, yakni tampilan daftar file dari aplikasi **File Explorer**.



PENJELASAN DARI SLIDE ke-08 s/d 19

9. Untuk mempermudah pengenalan terhadap objek **ListView** yang digunakan, umumnya seorang programmer memberikan penyingkatan nama terhadap **ListView** tersebut dengan penyingkatan nama "**lv_[nama]**", misalnya "**lvPegawai**", "**lvDonasi**", dan sebagainya.
10. Berikut penjelasan beberapa **Properties** yang umum digunakan pada **ListView** :

Properties	Keterangan
CheckBoxes	Mengatur kemunculan checkbox pada daftar item di dalam objek. Bernilai True atau False (Default = False).
Columns	Mengelola Kolom. Terdiri dari sejumlah method.
FullRowSelect	Mengatur bentuk visual pemilihan suatu daftar (highlight). Hanya berfungsi jika properties View = Details. Bernilai True atau False (Default = False).
GridLines	Mengatur kemunculan garis pemisah baris & kolom. Bernilai True atau False (Default = False).
Groups	Mengelola Grup (Kelompok). Hanya berfungsi jika properties ShowGroups = True. Terdiri dari sejumlah method.
HotTracking	Mengatur semua nilai dari semua kolom dari objek menjadi format HyperLink atau tidak. Bernilai True atau False (Default = False).
HoverSelection	Mengatur hanya nilai dari kolom pertama dari objek menjadi format HyperLink atau tidak. Bernilai True atau False (Default = False).
Items	Mengelola Daftar. Terdiri dari sejumlah method.
MultiSelect	Mengatur pemilihan daftar dapat lebih dari 1 (satu) daftar. Bernilai True atau False (Default = True).
ShowGroups	Mengelola kemunculan tampilan Grup (Kelompok). Bernilai True atau False (Default = True).
View	Mengelola bentuk tampilan daftar. Terdiri dari LargeIcon, Details, SmallIcon, List, dan Tile (Default = LargeIcon).

11. Berikut penjelasan beberapa **Method** yang umum digunakan pada **ListView** :

Method	Keterangan
CheckedItems	Untuk mengambil semua daftar yang memiliki tanda centang. Hanya berfungsi jika properties CheckBoxes = True.
Clear	Untuk menghapus semua nilai daftar dan kolom yang ada pada objek.
FindItemWithText	Untuk mencari suatu item dari daftar.

PENJELASAN DARI SLIDE ke-08 s/d 19

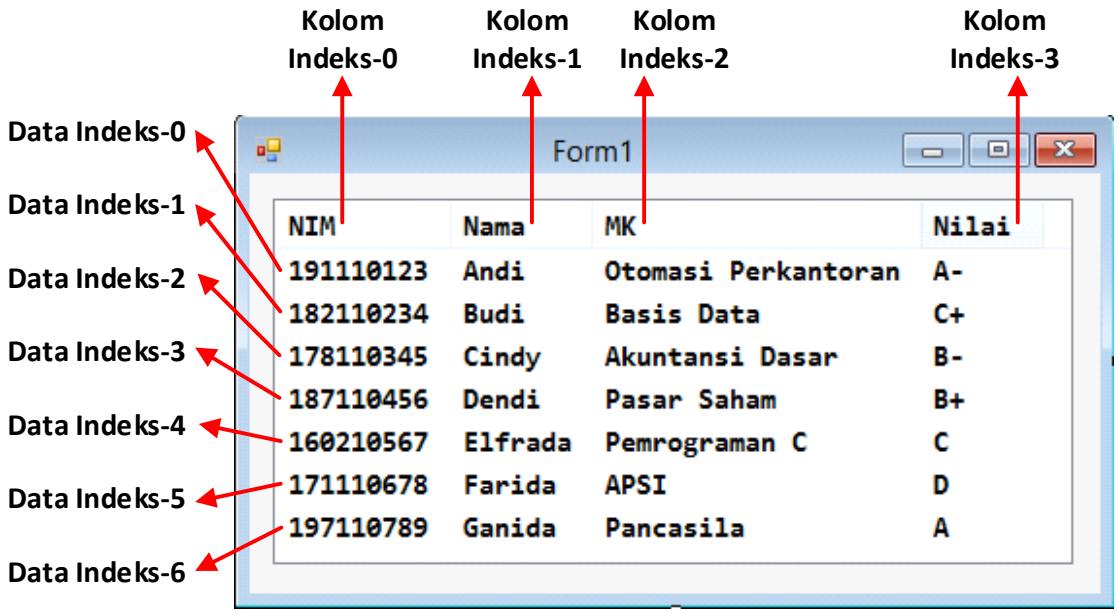
Method	Keterangan
FocusedItem	Untuk mengambil item yang sedang terpilih (highlight) dari daftar
SelectedItems	Untuk mengambil semua item yang sedang terpilih (highlight) dari daftar. Hanya berfungsi jika properties MultiSelect = True.
Columns.Add	Untuk menambahkan kolom baru.
Columns.Clear	Untuk menghapus semua kolom.
Columns.Count	Untuk mengambil jumlah kolom yang ada.
Columns.Insert	Untuk menyisipkan kolom baru pada posisi tertentu.
Columns.RemoveAt	Untuk menghapus kolom berdasarkan posisi indeks kolom.
Columns[index].Text	Untuk mengelola tulisan dari kolom indeks tertentu.
Columns[index].Width	Untuk mengelola lebar dari kolom indeks tertentu.
Groups.Add	Untuk menambahkan grup baru.
Groups.Clear	Untuk menghapus semua grup.
Groups.Count	Untuk mengambil jumlah grup yang ada.
Groups.Insert	Untuk menyisipkan grup baru pada posisi tertentu.
Groups.RemoveAt	Untuk menghapus grup berdasarkan posisi indeks grup.
Groups[index].Header	Untuk mengelola tulisan dari grup indeks tertentu.
Items.Add	Untuk menambahkan daftar baru.
Items.Clear	Untuk menghapus semua daftar.
Items.Count	Untuk mengambil jumlah daftar yang ada.
Items.IndexOf	Untuk melakukan pencarian daftar di dalam objek.
Items.Insert	Untuk menyisipkan daftar baru pada posisi tertentu.
Items.RemoveAt	Untuk menghapus daftar berdasarkan posisi indeks daftar.
Items[index].Checked	Untuk mengambil status checkbox. Hanya berfungsi jika properties CheckBoxes = True.
Items[index].Group	Untuk mengatur item pada indeks tertentu masuk ke dalam grup tertentu.
Items[index].Remove	Untuk menghapus item pada indeks tertentu.
Items[index].Selected	Untuk mengambil status apakah item indeks tertentu sedang terpilih (highlight) atau tidak.
Items[index].SubItems	Untuk mengelola Data Pendukung dari item indeks tertentu.
Items[index].Text	Untuk mengelola tulisan dari item indeks tertentu.

PENJELASAN DARI SLIDE ke-08 s/d 19

12. Berikut penjelasan beberapa **Event** yang umum digunakan pada **ListView** :

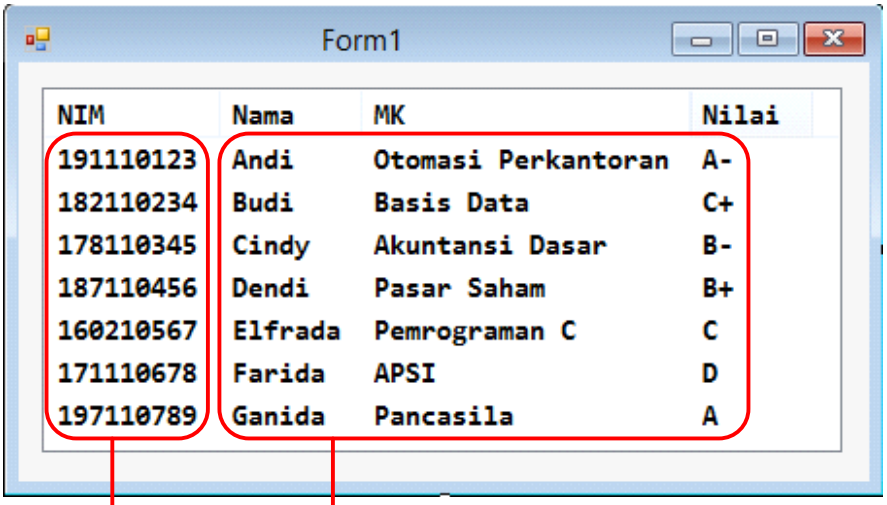
Event	Keterangan
ColumnClick	Dijalankan jika terjadi klik pada kolom.
ItemCheck	Dijalankan jika user mengklik checkbox pada suatu item. Hanya berfungsi jika properties CheckBoxes = True.
ItemChecked	Dijalankan jika user selesai mengklik checkbox pada suatu item. Hanya berfungsi jika properties CheckBoxes = True.

13. Dalam penggunaan **ListView**, semua item yang disajikan di dalam **ListView** memiliki konsep pengindeksan sebagai berikut :
    1. Setiap item (data) dari daftar (Items) di dalam **ListView** memiliki indeks, dimana data pertama berindeks-0, data ke-2 berindeks-1, data ke-3 berindeks-2, dan seterusnya (sama dengan konsep pengindeksan pada ListBox).
    2. Jika properties = Details, maka daftar yang disajikan di dalam **ListView** akan berbentuk Tabel, sehingga memiliki Kolom (Columns). Kolom di dalam **ListView** memiliki indeks, dimana kolom pertama berindeks-0, kolom ke-2 berindeks-1, kolom ke-3 berindeks-2, dan seterusnya.



    3. 1 item (data) di dalam **ListView** dapat terdiri dari beberapa sub data (sub-item) yang dinyatakan dalam beberapa kolom, dimana data yang tersimpan pada kolom indeks ke-0 (pertama) dinamakan sebagai **Data Utama**, sedangkan data yang tersimpan pada kolom indeks ke-1 (kedua), dan seterusnya dinamakan sebagai **Data Pendukung**.

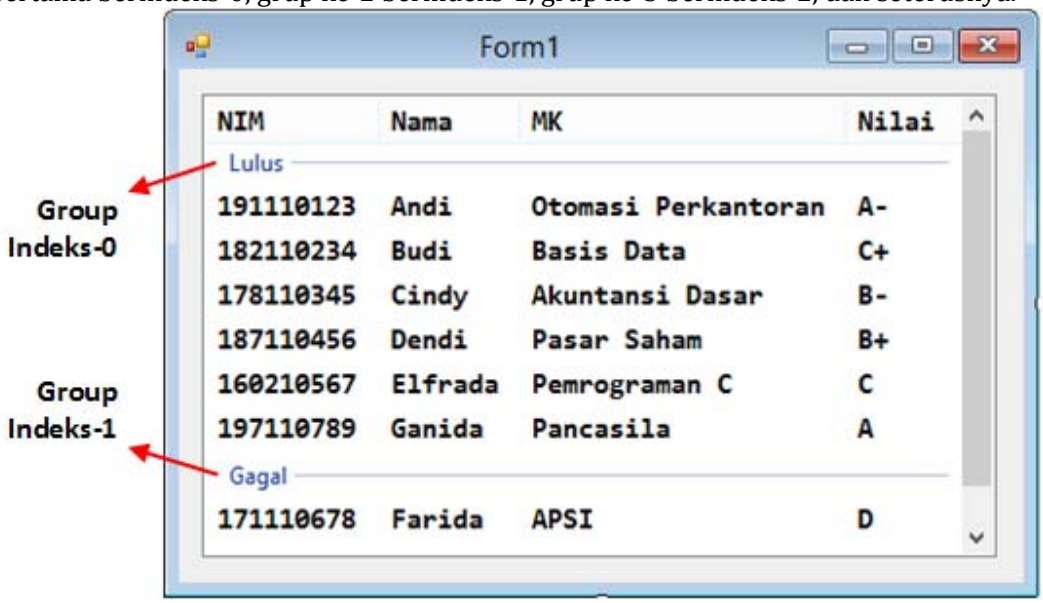
PENJELASAN DARI SLIDE ke-08 s/d 19



Kolom indeks-0 (Data Utama)

Kolom indeks-1 dan seterusnya (Data Pendukung)
Dari contoh :
- Nama dari pemilik NIM
- MK yang diambil oleh pemilik NIM
- Nilai dari MK yang diambil oleh pemilik NIM

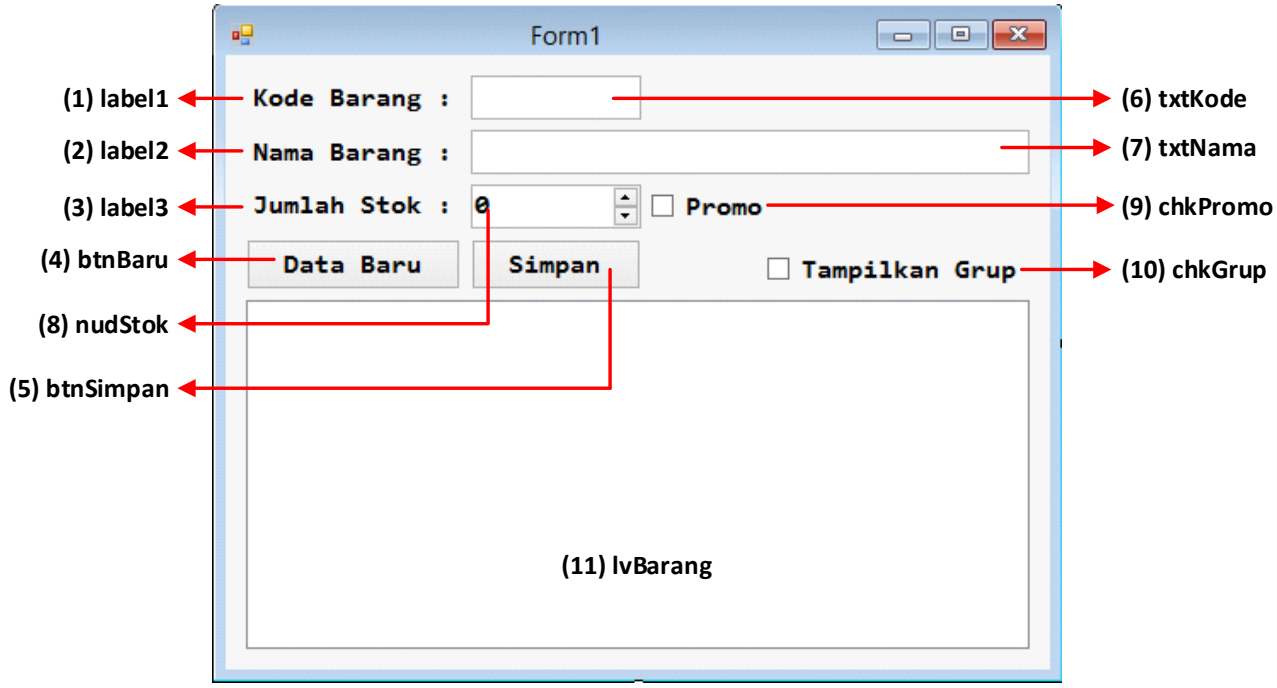
4. Penyajian item di dalam **ListView**, dapat dinyatakan kategori ke dalam Grup (Groups), dimana grup pertama berindeks-0, grup ke-2 berindeks-1, grup ke-3 berindeks-2, dan seterusnya.



CONTOH KASUS SEDERHANA - 2

Untuk contoh kasus sederhana dari penggunaan objek **ListView**, dapat dilihat contoh berikut :

1. Rancang tampilan Windows Form Application sebagai berikut :



2. Untuk masing-masing objek dari rancangan tampilan di atas, memiliki keterangan **Properties** sebagai berikut :

   a. Untuk **Properties** dari **Form** :

Properties	Keterangan
Font	Name = Consolas Size = 12 Style = Bold

   b. Untuk **Properties** dari **Label** "**label1**" **(1)** :

Properties	Keterangan
Name	"label1"
Text	"Kode Barang :"

   c. Untuk **Properties** dari **Label** "**label2**" **(2)** :

Properties	Keterangan
Name	"label2"
Text	"Nama Barang :"

CONTOH KASUS SEDERHANA - 2

d. Untuk **Properties** dari **Label** "**label3**" **(3)** :

Properties	Keterangan
Name	"label3"
Text	"Jumlah Stok :"

e. Untuk **Properties** dari **Button** "**btnBaru**" **(4)** :

Properties	Keterangan
Name	"btnBaru"
Text	"Data Baru"

f. Untuk **Properties** dari **Button** "**btnSimpan**" **(5)** :

Properties	Keterangan
Name	"btnSimpan"
Text	"Simpan"

g. Untuk **Properties** dari **TextBox** "**txtKode**" **(6)** :

Properties	Keterangan
Name	"txtKode"
Text	"" <Kosongkan>

h. Untuk **Properties** dari **TextBox** "**txtNama**" **(7)** :

Properties	Keterangan
Name	"txtNama"
Text	"" <Kosongkan>

i. Untuk **Properties** dari **NumericUpDown** "**nudStok**" **(8)** :

Properties	Keterangan
Name	"nudStok"

j. Untuk **Properties** dari **CheckBox** "**chkPromo**" **(9)** :

Properties	Keterangan
Name	"chkPromo"

k. Untuk **Properties** dari **CheckBox** "**chkGrup**" **(10)** :

Properties	Keterangan
Name	"chkGrup"

l. Untuk **Properties** dari **ListView** "**lvBarang**" **(11)** :

Properties	Keterangan
Name	"lvBarang"

CONTOH KASUS SEDERHANA - 2

3. Untuk masing-masing objek dari rancangan tampilan di atas, memiliki **Event** dengan kode program sebagai berikut :

   a. **Event** "**Load**" dari **Form** :

```
nudStok.Minimum = 1;
nudStok.Maximum = 9999;
chkGrup.Checked = true;
lvBarang.Columns.Add("Kode", 100);
lvBarang.Columns.Add("Nama", 280);
lvBarang.Columns.Add("Stok", 50, HorizontalAlignment.Center);
lvBarang.Groups.Add(new ListViewGroup("Promo"));
lvBarang.Groups.Add(new ListViewGroup("Reguler"));
lvBarang.View = View.Details;
lvBarang.GridLines = true;
```

   Penjelasan untuk kode program di atas adalah :

   - Untuk potongan kode program :

```
nudStok.Minimum = 1;
nudStok.Maximum = 9999;
chkGrup.Checked = true;
```

     Menset batasan nilai yang dapat dipilih pada **NumericUpDown** "**nudStok**" adalah antara 1 s/d 9999, serta status centang awal dari **CheckBox** "**chkGrup**" adalah tercentang.

   - Untuk potongan kode program :

```
lvBarang.Columns.Add("Kode", 100);
lvBarang.Columns.Add("Nama", 280);
lvBarang.Columns.Add("Stok", 50, HorizontalAlignment.Center);
```

     Menambahkan 3 (tiga) kolom ke dalam **ListView** "**lvBarang**", yakni kolom "**Kode**" (berindeks 0) dengan lebar kolom secara default sebesar 100 piksel, kolom "**Nama**" (berindeks 1) dengan lebar kolom secara default sebesar 280 piksel, dan kolom "**Stok**" (berindeks 2) dengan lebar kolom secara default sebesar 50 piksel dan perataan tengah.,

   - Untuk potongan kode program :

```
lvBarang.Groups.Add(new ListViewGroup("Promo"));
lvBarang.Groups.Add(new ListViewGroup("Reguler"));
```

     Menambahkan 2 (dua) grup ke dalam **ListView** "**lvBarang**", yakni grup "**Promo**" (berindeks 0), dan grup "**Reguler**" (berindeks 1).

   - Untuk potongan kode program :

```
lvBarang.View = View.Details;
lvBarang.GridLines = true;
```

     Menset bentuk tampilan dari **ListView** "**lvBarang**" adalah bentuk Details, serta garis pemisah baris & kolom dimunculkan.

   b. **Event** "**Click**" dari **Button** "**btnBaru**" :

```
txtKode.Clear();
txtNama.Clear();
nudStok.Value = 1;
chkPromo.Checked = false;
txtKode.Focus();
```

CONTOH KASUS SEDERHANA - 2

Penjelasan untuk kode program di atas adalah :
Mengosongkan **TextBox** "**txtKode**" dan "**txtNama**",
Mengatur kembali nilai awal **NumericUpDown** "**nudStok**" menjadi 1,
Menghilangkan tanda centang dari **CheckBox** "**chkPromo**", dan
Memberikan fokus kursor ke **TextBox** "**txtKode**".

c. **Event** "**Click**" dari **Button** "**btnSimpan**" :

```
lvBarang.Items.Add(txtKode.Text);
lvBarang.Items[lvBarang.Items.Count - 1].SubItems.Add(txtNama.Text);
lvBarang.Items[lvBarang.Items.Count - 1].SubItems.Add(
  nudStok.Value.ToString());
if (chkPromo.Checked)
  lvBarang.Items[lvBarang.Items.Count - 1].Group = lvBarang.Groups[0];
else
  lvBarang.Items[lvBarang.Items.Count - 1].Group = lvBarang.Groups[1];
btnBaru_Click(null, null);
```

Penjelasan untuk kode program di atas adalah :

- Untuk potongan kode program :

```
lvBarang.Items.Add(txtKode.Text);
```

Memasukkan nilai properties Text dari **TextBox** "**txtKode**" ke dalam **ListView** "**lvBarang**" pada baris baru & kolom pertama (kolom "**Kode**").
Setelah nilai tersebut masuk ke dalam **ListView** "**lvBarang**", maka method **.Items.Count()** dari **ListView** "**lvBarang**" secara auto akan diincrement dengan 1, yang artinya telah bertambah 1 (satu) baris data pada **ListView** "**lvBarang**".

- Untuk potongan kode program :

```
lvBarang.Items[lvBarang.Items.Count - 1].SubItems.Add(txtNama.Text);
lvBarang.Items[lvBarang.Items.Count - 1].SubItems.Add(
  nudStok.Value.ToString());
```

Memasukkan nilai properties Text dari **TextBox** "**txtNama**" ke dalam **ListView** "**lvBarang**" pada baris terakhir & kolom ke-2 (kolom "**Nama**").
Memasukkan nilai properties Value dari **NumericUpDown** "**nudStok**" yang telah dikonversi menjadi tipe String ke dalam **ListView** "**lvBarang**" pada baris terakhir & kolom ke-3 (kolom "**Stok**").

Note : Proses memasukkan nilai ke dalam kolom ke-2 dan seterusnya (jenis kolom data pendukung) adalah bersistem sequential (berurutan), sehingga apabila 2 baris dari contoh di atas dibalikkan, maka nilai inputan Nama akan tersimpan ke kolom "**Stok**", dan nilai inputan Stok akan tersimpan ke kolom "**Nama**".

- Untuk potongan kode program :

```
if (chkPromo.Checked)
  lvBarang.Items[lvBarang.Items.Count-1].Group = lvBarang.Groups[0];
else
  lvBarang.Items[lvBarang.Items.Count-1].Group = lvBarang.Groups[1];
btnBaru_Click(null, null);
```

CONTOH KASUS SEDERHANA - 2

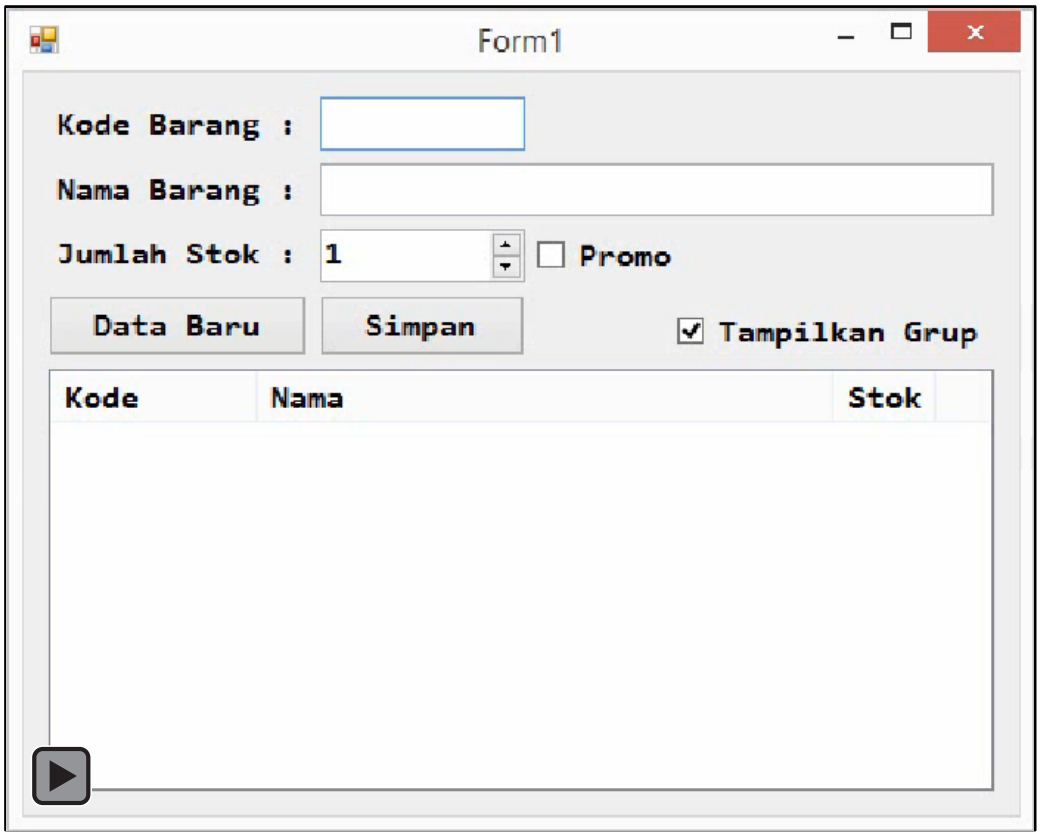
Jika **CheckBox** "**chkPromo**" dicentang, maka :
Data pada baris terakhir akan di set ke grup berindeks 0, yakni grup "**Promo**".
Jika tidak, maka :
Data pada baris terakhir akan di set ke grup berindeks 1, yakni grup "**Reguler**".

d. **Event** "**CheckedChanged**" dari **CheckBox** "**chkGrup**" :

```
lvBarang.ShowGroups = chkGrup.Checked;
```

Penjelasan untuk kode program di atas adalah :
Nilai properties ShowGroups dari **ListView** "**lvBarang**" akan bernilai sama dengan nilai status centang dari properties Checked dari **CheckBox** "**chkGrup**", sehingga jika status centang pada **CheckBox** "**chkGrup**" adalah True, maka properties ShowGroups dari **ListView** "**lvBarang**" juga akan bernilai True, maka grup akan ditampilkan ke **ListView** "**lvBarang**".
Sebaliknya jika status centang pada **CheckBox** "**chkGrup**" adalah False, maka properties ShowGroups dari **ListView** "**lvBarang**" juga akan bernilai False, maka grup akan disembunyikan dari tampilan **ListView** "**lvBarang**".

4. Dari contoh kasus sederhana ini, jika setelah selesai dirancang dan dituliskan kode program untuk **Event** yang ada di atas, maka akan memiliki tampilan video sebagai berikut :



Note : Memerlukan Adobe Flash Player untuk dapat memutar video di atas.